A Privacy-Preserving Target Time Management System for Air Traffic Flow Management

Sebastian Gruber*, Tobias Harzfeld†, Christoph G. Schuetz‡,
Christoph Fabianek§, Christoph Rihacek¶, Eduard Gringinger‖,
Florian Wohner**, and Thomas Loruenser††
*Johannes Kepler University Linz, Linz, Austria
ORCID: 0000-0002-5714-1870
†Johannes Kepler University Linz, Linz, Austria
ORCID: 0009-0003-0275-6207
‡Johannes Kepler University Linz, Linz, Austria
ORCID: 0000-0002-0955-8647
§Frequentis AG, Vienna, Austria
ORCID: 0009-0002-4410-8796
¶Frequentis AG, Vienna, Austria
christoph.rihacek@frequentis.com
‖Frequentis AG, Vienna, Austria
ORCID: 0000-0003-3897-3003
**AIT Austrian Institute of Technology, Vienna, Austria
ORCID: 0000-0002-8641-7522
††AIT Austrian Institute of Technology, Vienna, Austria
ORCID: 0000-0002-1829-4882

***Abstract*—In Air Traffic Flow Management (ATFM), flights are often assigned new arrival times in case of unexpected events such as poor weather conditions. The assignment of arrival times results in flight delays and thus additional costs for airspace users (AUs) and reduced efficiency for the airport. Since the impact of delays differs across flights, the airport and AUs would prefer to prioritize important flights. For this purpose, a Target Time Management System (TTMS) has been proposed that enables AUs and the airport to collaboratively prioritize flights and optimize flight lists. In this paper, we evaluate an implementation of a privacy-preserving TTMS that combines evolutionary algorithms and secure multi-party computation (MPC) to protect the confidentiality of AUs' preferences. We use data from 51 real-world regulation shared by Zurich Airport for the experimental evaluation. The results indicate that a privacy-preserving TTMS can find solutions almost as good as the solutions found by a deterministic algorithm, while protecting the preferences of AUs. In addition, runtime measurements are reported to demonstrate that the TTMS is capable of finding solutions within practical time constraints. The privacy-preserving TTMS is a promising alternative for settings in which AUs do not fully trust the platform provider.**

***Index Terms*—flight prioritization, privacy-preserving optimization, multi-party computation, evolutionary algorithm**

I. Introduction

In the European air traffic network, the EUROCONTROL Network Manager regulates arrivals of flights at certain airports when demand exceeds airport capacity. In such cases, each regulated flight is assigned a new target time of arrival (TTA) at the regulated airport, leading to uniformly delayed arrival times of flights when reorganizing flight lists on a first-planned, first-served basis. The resulting flight delays lead to additional costs for the airspace users (AUs), e.g., due to missing passenger connections, necessary crew replacements, and reputational damage, but also for the airport, e.g., due to reduced efficiency of airport operations.

In practice, not all flights have the same criticality for AUs and airport, with the costs of arrival delay varying between flights. Consequently, in case of a regulation, AUs would prefer to prioritize the more critical flights to reduce delay costs, while the arrival airport would prioritize flights in a way that streamlines airport operations. To this end, a Target Time Management System (TTMS) has been proposed as a solution, which would enable flight prioritization across AUs. Such a TTMS would receive requests for a TTA for each flight, from both AUs and the airport, as inputs for an optimization of the flight list. Zurich Airport, for example, has already conducted live trials with such a TTMS [1].

The TTMS implemented at Zurich Airport collects inputs from airspace users and the airport to conduct a deterministic optimization of the assignment problem, assigning TTAs to flights. Although the TTMS was successfully tested, and the results indicate increased arrival punctuality and fewer missed connections [1], the TTMS developed at Zurich Airport assumes a fully trusted platform provider, which may not be the case at other airports, where AUs may be more reluctant to share confidential information about flight importance with a platform provider.

In this paper, we investigate the practicability of extending the TTMS concept for scenarios where, unlike the situation at Zurich Airport, the platform provider optimizing the flight lists is not fully trusted. Hence, we consider a privacy-preserving TTMS, hosted by an honest-but-curious platform provider, using secure multi-party computation (MPC) in combination with an evolutionary algorithm to nevertheless protect the confidential preferences of AUs from unauthorized access and from the platform provider. In a controlled laboratory exercise, with real-world data shared by Zurich Airport from their TTMS live trials [1], we therefore investigated the performance of the proposed privacy-preserving TTMS. While MPC introduces computational overhead that may result in a deterministic algorithm not being able to finish within the time available for optimization, the combination with an evolutionary algorithm promises to be able to find solutions of sufficient quality within the required time limit.

In our laboratory exercise with the privacy-preserving TTMS, we used data from 51 real-world regulations shared by Zurich Airport to measure the performance of using an evolutionary algorithm in combination with MPC when optimizing flight lists. We calculated a subset of Pareto-optimal solutions for each flight list that was optimized during the 51 regulations. The evolutionary algorithm was then executed on each flight list with different configurations, and the optimization results were compared with the corresponding subset of Pareto-optimal solutions. Our results show that the evolutionary algorithm is able to find solutions that are almost as good as the optimal solutions found by the deterministic algorithm in the live trials at Zurich Airport. These results indicate that using an evolutionary algorithm in combination with MPC has negligible impact on the quality of the flight lists found by the TTMS. A privacy-preserving implementation of a TTMS is thus a viable alternative for settings where a platform provider may not be fully trusted by AUs.

The remainder of this paper is structured as follows. Section II provides background on collaborative flight prioritization. Section III introduces the privacy-preserving TTMS. Section IV describes the experimental setup. Section V presents the results. Section VI concludes the paper.

II. Background and Related Work

In Air Traffic Flow Management (ATFM), flights are regulated when arrival demand exceeds airport capacity, for example due to poor weather conditions. To smooth arrival demand, the EUROCONTROL Network Manager reduces the arrival rate and assigns TTAs to flights on a first-planned, first-served basis. This assignment of TTAs results in flight delays and additional costs for AUs and reduced efficiency for the airport. The impact of delay on additional costs and efficiency varies between flights, and, therefore, the airport and AUs would prefer to prioritize more important flights in order to improve overall efficiency.

The User-Driven Prioritization Process (UDPP) [2] improves flexibility of AUs during ATFM regulations. UDPP allows each AU to prioritize its own flights while ensuring that flights of other AUs are not negatively affected. Validation results demonstrate that UDDP reduces operational costs of AUs and improves passenger connections [3].

The SlotMachine project [4] developed a platform for collaborative flight prioritization to further improve the flexibility of AUs during ATFM regulations. In case of an ATFM regulation, AUs who want to participate in optimization provide preferences for their flights. The preferences for a flight are encoded as a weight map, consisting of numerical values (i.e, weights) for each available TTA. A weight represents the utility of a particular flight arriving at a particular TTA. Collaborative flight prioritization is transformed into an assignment problem with the aim of maximizing the utility of assigning flights to TTAs. AUs may not fully trust the platform provider and only participate in optimization if their preferences remain confidential. Privacy-preserving implementations of deterministic algorithms may not be able to finish within a deadline, and therefore, the TTMS combines genetic algorithms and MPC for privacy-preserving optimization [5]. Market mechanisms were introduced based on real-world currency [6] and a virtual delay credit [7] to ensure fairness and equity among AUs.

The HARMONIC project [8] extended the TTMS by allowing the airport to also provide preferences. Collaborative flight prioritization thus became a multi-objective assignment problem, where the airport provides preferences for each flight. The project investigated two settings: One in which the platform provider is trusted and another in which the platform provider is not fully trusted [9]. In the former, a deterministic algorithm provides flight lists that are Pareto-optimal with respect to the weight maps of the airport and the AUs [1]. In the latter, evolutionary algorithms for multi-objective optimization are combined with MPC to optimize flight lists [10]. Instead of market mechanisms, conformance criteria for the inputs of AUs [11] are introduced and equity mechanisms that modify the weights of AUs are investigated based on already existing concepts [12].

III. Privacy-Preserving Flight Prioritization

In this paper, we explore the practicability of a privacy-preserving TTMS for optimizing flight lists. Privacy-preserving implementations of deterministic algorithms [13] may not be able to finish within a deadline, especially when a larger number of flights and TTAs are involved in the ATFM regulation. Existing approaches for privacy-preserving optimization using evolutionary algorithms either protect the whole evolutionary algorithm [14], [15] or only the objective function [16], [17]. The former suffers from a stronger impact of privacy-preserving computations on runtime, while the latter reveals information about the objective function. We adopt the approach developed in the SlotMachine [4] and HARMONIC [8] projects, which reduces the impact of privacy-preserving computations on runtime while reducing the information that is revealed about the objective function.

Figure 1 illustrates the architecture of a privacy-preserving TTMS. The airport and AUs continuously submit their preferences for each flight and TTA to the Orchestrator. Each

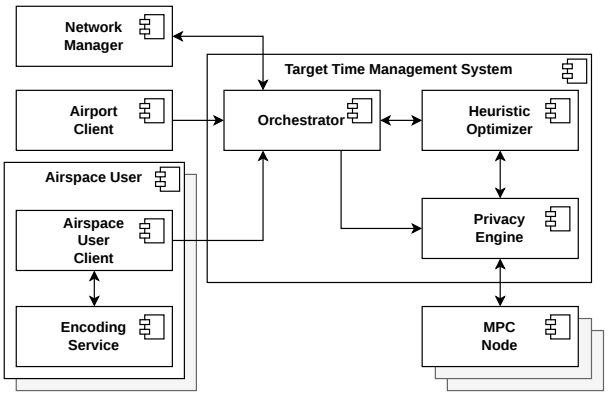

Fig. 1. Privacy-preserving target time management system, adapted from [9]

AU operates a local Encoding Service that encodes and encrypts the preferences using the public keys of the MPC nodes before transmitting them to the Orchestrator. When a regulation is announced, the Orchestrator receives input from the Network Manager and initiates the optimization process. The Orchestrator therefore forwards public information (i.e., available TTAs and flights) together with the airport preferences to the Optimizer, while encrypted AU preferences are sent to the Privacy Engine, which distributes them among the MPC nodes. If an AU did not provide valid preferences, the Orchestrator uses default preferences for the corresponding flights. The Optimizer executes an evolutionary algorithm that in combination with the Privacy Engine optimizes flight lists. When a termination criterion is met, the Optimizer returns the optimization result to the Orchestrator, which selects a flight list and submits a request to the Network Manager.

The Optimizer produces populations of solutions, while the Privacy Engine evaluates these populations and provides the evaluation results to the Optimizer to further guide the search process. The privacy-preserving TTMS reduces the information revealed to an honest-but-curious platform provider by applying obfuscation methods. These methods obfuscate the evaluation results received by the Privacy Engine and the Optimizer. The following obfuscation methods have been proposed [5]: *Order* obfuscation reveals only the ordering of solutions instead of their fitness values, *Above Threshold* obfuscation reveals the solutions whose fitness values exceed a configurable threshold, *Top Individuals* obfuscation reveals a configurable number of top-performing solutions based on their fitness values, *Fitness Buckets* obfuscation distributes solutions into buckets, where each bucket covers a range of fitness values, and reveals only the bucket assignment, *Order Quantiles* obfuscation distributes solutions into quantiles, where each quantile contains approximately the same number of solutions, and reveals only the quantile assignment.

The questions arises whether a privacy-preserving TTMS can find flight lists of sufficient quality and in an appropriate amount of time. Previous studies [5], [10], [18] evaluated the performance of this approach using synthetic datasets. We experimentally evaluate the performance of the privacy-preserving TTMS in terms of solution quality and optimization runtime using data from 51 real-world regulations shared by Zurich Airport.

IV. Experimental Setup

In this section, we describe the setup used for the experimental evaluation of the privacy-preserving TTMS. The objective of the experiments is to demonstrate the impact of using evolutionary algorithms in combination with MPC on the quality of flight lists and runtime.

A. Prototypes

Our implementation of the Optimizer [19] uses the deterministic algorithm provided by SciPy [20] (version 1.14.1) to solve the assignment problem and PyGAD [21] (version 3.4.0) to run genetic algorithms. We modified PyGAD's implementation of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [22] to improve its runtime behavior. Instead of sorting the entire population before selecting parents, the modified NSGA-II implementation terminates non-dominated sorting when the configured number of parents has been selected. We further added the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [23], and two variants of a simple multi-objective tournament selection. One tournament selector selects a random non-dominated individual from k random individuals (i.e., the tournament size). The other tournament selector selects k random individuals and repeatedly pairs participants, selecting winners based on Pareto dominance until only one individual remains. We also extended PyGAD with shift mutation and partially matched crossover.

The Optimizer maintains a set of estimated non-dominated solutions during an optimization to preserve the best solutions found so far. We refer to this solution set as the *optimization result*. These solutions are non-dominated with respect to their actual fitness for the non-obfuscated objective and their estimated fitness for the obfuscated objective. The Optimizer continuously updates the optimization result during optimization by adding the estimated non-dominated solutions of the current population and retaining only those that remain non-dominated. The performance metrics are computed from the final optimization results.

In our experimental setup, we simulate the Orchestrator who provides an instance of an assignment problem and a configuration for either the deterministic algorithm or a genetic algorithm as input for the Optimizer. The Optimizer receives weight maps as plain text and submits the weights to the Encoding Service of the Privacy Engine. After receiving the encoded weights, the Optimizer forwards them to the Privacy Engine for distribution across the MPC nodes.

Our implementation of the Privacy Engine [24] uses MP-SPDZ [25] for MPC. The Privacy Engine consists of a Controller, a Encoding Service and three MPC nodes. The Encoding Service encodes the input weights, and returns the encoded weights to the Optimizer. The Optimizer provides the encoded weights to the Controller, which distributes them

among the MPC nodes. When the Optimizer provides a population of candidate solutions to the Controller of the Privacy Engine, the Controller initiates a fitness evaluation using the MPC nodes. We employ three MPC nodes, and assume that at least two MPC nodes behave honestly. Therefore, we can use a more lightweight protocol, such as the rep-field protocol [26]. The Controller receives an obfuscated evaluation result from the MPC nodes, and returns the result to the Optimizer. The implementation of the Privacy Engine and runtime benchmarks are available online [24].

The Optimizer includes a simulation of the Privacy Engine to run optimizations with obfuscation while minimizing the impact on runtime. The result of an optimization is identical when using the simulation of the Privacy Engine and our implementation of the Privacy Engine for evaluation.

B. Datasets

We use synthetic datasets for parameter tuning and real-world data for performance evaluation. For parameter tuning, we generated five instances of the bi-objective assignment problem with a problem size of 100×100 and random weight map values.

For performance evaluation, we received weight maps from 51 regulations that were collected during an eight-week trial of the TTMS with a fully trusted platform provider at Zurich Airport [1]. Within a regulation, optimizations were continuously performed using the remaining flights and TTAs. In total, we received 1 017 instances across all regulations, excluding instances with a problem size of 1×1. These 1 017 instances are primarily imbalanced assignment problems, with more TTAs than flights.

For calculating the performance metrics, we identified a subset of Pareto-optimal solutions for each instance. We therefore applied the deterministic algorithm of the Optimizer to 1 001 differently weighted combinations of the two objectives with the aim to maximize the assignment sum. We additionally identified the worst solutions by applying the deterministic algorithm to differently weighted combinations of the two objectives with the aim to minimize the assignment sum. The minimum fitness values (i.e., assignment sums) tend to be several orders of magnitude lower than the maximum, since infeasible assignments are penalized with large negative weights, whereas feasible assignments have positive weights.

C. Metrics

We measure the performance of a solution set using the Generational Distance (GD) and Inverted Generational Distance (IGD). Let S be a set of solutions and P the subset of Pareto-optimal solutions, which we previously identified using the deterministic algorithm. GD measures the average distance from each solution in S to the nearest solution in P, whereas IGD measures the average distance from each solution in P to the nearest solution in S [27]. Small values for GD indicate good convergence, and small values for IGD indicate good diversity [27]. We modified the distance calculation for GD and IGD to take into account the dominance relation between two solutions [28]. We therefore refer to these metrics as GD^+ and IGD^+ [28].

Before calculating GD^+ and IGD^+ for performance evaluation, we normalized the fitness values of the solutions in the optimization results. The fitness of a solution for objective i is normalized based on the known maximum and minimum fitness values for objective i, whereas a normalized fitness value of 0 corresponds to the minimum (i.e., worst) fitness and 1 to the maximum (i.e., ideal) fitness.

D. Experiments

We performed three experiments to determine a final configuration of the genetic algorithm for each obfuscation configuration and two experiments for performance evaluation. In the experiments, we used *above threshold* obfuscation with a threshold of 90 %, *fitness buckets* obfuscation with 10 buckets, *order* obfuscation, *order quantiles* obfuscation with 10 quantiles, and *top individuals* obfuscation returning the top 10 % of the individuals of a population. We used these configurations to obfuscate one of the two objectives in Experiment 1-3 and the AU objective in Experiment 4-5.

In the first experiment, we executed a meta-genetic algorithm for each combination of obfuscation configuration and either algorithm (i.e., NSGA-II and SPEA2) or parent selection type (i.e., the two variants of multi-objective tournament selection), resulting in 20 different executions of the meta-genetic algorithm. The individuals of a meta-genetic algorithm are parameter sets for the configuration of a genetic algorithm. The meta-genetic algorithm evaluates a parameter set by configuring and executing the genetic algorithm with these parameters on Instance 1 and Instance 2 of the generated bi-objective assignment problems. The fitness of a parameter set are the GD^+ values obtained from the optimization results for both instances. The meta-genetic algorithm was configured with NSGA-II, 40 generations, a population size of 50, 25 parents, random mutation with a probability of 10 %, single point crossover, and keeping one elitist parameter set.

The options for the configuration of each genetic algorithm are shown in Table I. The number of generations is calculated as 100 000 divided by the population size to maintain a comparable number of evaluations across configurations. The crossover procedure depends on the crossover probability: If a crossover probability is provided, parents are selected for recombination based on that probability. If no crossover probability is provided, parents are recombined pairwise. The archive size is an additional configuration option for SPEA2 and the tournament size for the two variants of the multi-objective tournament selection.

For the second experiment, we selected at least 20 of the best-performing parameter sets found by each executed meta-genetic algorithm. We therefore performed non-dominated sorting of the parameter sets using their GD^+ values. For each meta-genetic algorithm, all parameter sets from the best Pareto fronts were selected, starting with the first front, until at least 20 parameter sets were obtained.

TABLE I
OPTIONS FOR CONFIGURING A GENETIC ALGORITHM. PMX = PARTIALLY MATCHED CROSSOVER; SP = SINGLE POINT; TP = TWO POINTS.

Parameter	Options
Population Size	{100, 300, 500}
Number of Parents	{0.1, 0.2, 0.3, 0.4, 0.5} × Population Size
Crossover Type	{PMX, SP, TP, Uniform, Scattered}
Crossover Probability	{0, 20, 40, 60, 80, 100} % (optional)
Mutation Type	{Swap, Inversion, Scramble, Shift}
Mutated Genes	{5, 10, 15, 20, 25, 30, 40, 50} %
Archive Size	{0.1, 0.2, 0.3, 0.4, 0.5} × Population Size
Tournament Size	{0.02, 0.04, 0.06, 0.08} × Population Size

In the second experiment, we configured and executed the genetic algorithm ten times with each of the selected best parameter sets from Experiment 1 on Instance 3 and Instance 4 of the generated bi-objective assignment problems.

For the third experiment, we selected the best parameter set for each combination of obfuscation configuration and either algorithm or parent selection type. We therefore performed non-dominated sorting of the runs using their GD^+ values on Instance 3 and Instance 4. For each parameter set, we computed the median and mode of the Pareto front ranks across its runs. The best parameter set for each combination of obfuscation configuration and either algorithm or parent selection type was selected based on the lowest median rank. For tie-breaking, we used the mode rank.

In the third experiment, we configured and executed the genetic algorithm 21 times with the selected best parameter sets from Experiment 2 on Instance 5 of the generated bi-objective assignment problems.

For performance evaluation, we determined a single final configuration of the genetic algorithm for each obfuscation configuration. Therefore, for each algorithm or parent selection type, we selected the run with the median GD^+ value. Except for *above threshold* obfuscation, the median GD^+ runs of NSGA-II achieved lower GD^+ values than those of SPEA2 and the multi-objective tournament selectors after approximately 100 000 evaluations. We therefore selected, for each obfuscation configuration, the configuration with NSGA-II as the final configuration. These configurations are shown in Table II.

The fourth and fifth experiments were conducted to evaluate the performance of genetic algorithms with different obfuscation configurations on instances from 51 regulations. In the fourth experiment, we configured the genetic algorithm with the configurations listed in Table II and executed it on all 1 017 instances. In the first four experiments, the Privacy Engine was simulated by the Optimizer to reduce the runtime of the experiments. In the fifth experiment, we used our implementation of the Privacy Engine and optimized the first instance of each regulation. The optimization results obtained in the fifth experiment are consistent with those from the fourth experiment for the corresponding instances. We used the fifth experiment as a proof-of-concept as well as to measure the run time of optimization runs with the Privacy Engine.

V. EXPERIMENTAL RESULTS

In this section, we present the results of the performance evaluation using real-world instances from 51 regulations. The objective of the experiments is to demonstrate the impact of using evolutionary algorithms in combination with MPC on the quality of flight lists and runtime.

Figure 2 presents optimization results for a sample scenario comprising 122 flights and 233 TTAs. The figure illustrates the corresponding Pareto-optimal subset, which appears relatively dense. This may indicate a similarity between the default weight maps constructed by the airport for AUs that do not submit preferences and the airport's weight map. While the performance of the evolutionary algorithm varies across configurations, the configurations also differ in the amount of information revealed, and the results are therefore not directly comparable.

Table III and Table IV report the median and interquartile range (IQR) of GD^+ and IGD^+, respectively, of the optimization results obtained by the genetic algorithm for each obfuscation configuration. These values are also reported for the estimated non-dominated solutions from the initial populations (i.e., random solutions) to provide a baseline. The instances are grouped into five equal-width bins with a varying number of instances across bins based on problem size, i.e., the number of TTAs and flights. Median and IQR values of GD^+ reported in Table III are similar to those of IGD^+ in Table IV. This is due to relatively dense sets of Pareto-optimal solutions rather than a widely diverse one, indicating a positive correlation between airspace user preferences and airport preferences.

The performance of initial populations already appears acceptable considering that fitness values for each objective are normalized to the range between 0 and 1, and that lower GD^+ and IGD^+ values indicate better solution sets. However, the minimum fitness values used for normalization are several orders of magnitude smaller than the maximum fitness values. Consequently, although the results may appear acceptable at first glance, the performance of the initial populations is in fact insufficient. For instances with problem sizes between 3 and 73, the performance still appears strong. This range contains several small instances, such as instances of size 1×2 with only two feasible solutions. The initial population is therefore likely to already contain Pareto-optimal solutions for a subset of instances in this problem size range.

The optimization results of genetic algorithms exhibit substantially better performance than the initial populations. In particular, the median and IQR of GD^+ and IGD^+ are several orders of magnitude smaller than those of the initial populations. As the problem size increases, the median GD^+ and IGD^+ values also increase, indicating a reduction in performance. Large instances have a greater number of feasible solutions, making optimization more challenging.

Table V reports minimum, median, maximum and IQR of optimization runtimes for different problem sizes when using the Privacy Engine and assuming no network latency. We grouped the instances into three equal-width bins with a

TABLE II
CONFIGURATIONS OF THE GENETIC ALGORITHM FOR PERFORMANCE EVALUATION

Parameter	Above 90 % Threshold	10 Fitness Buckets	Order	10 Order Quantiles	Top 10 % Individuals
Parent Selection Type	NSGA-II	NSGA-II	NSGA-II	NSGA-II	NSGA-II
Population Size	300	500	500	500	500
Number of Generations	333	200	200	200	200
Number of Parents	30	50	50	50	50
Crossover Type	Uniform	Uniform	Scattered	Uniform	Scattered
Crossover Probability	60 %	40 %	80 %	80 %	-
Mutation Type	Swap	Swap	Swap	Swap	Swap
Mutated Genes	10 %	5 %	10 %	20 %	15 %

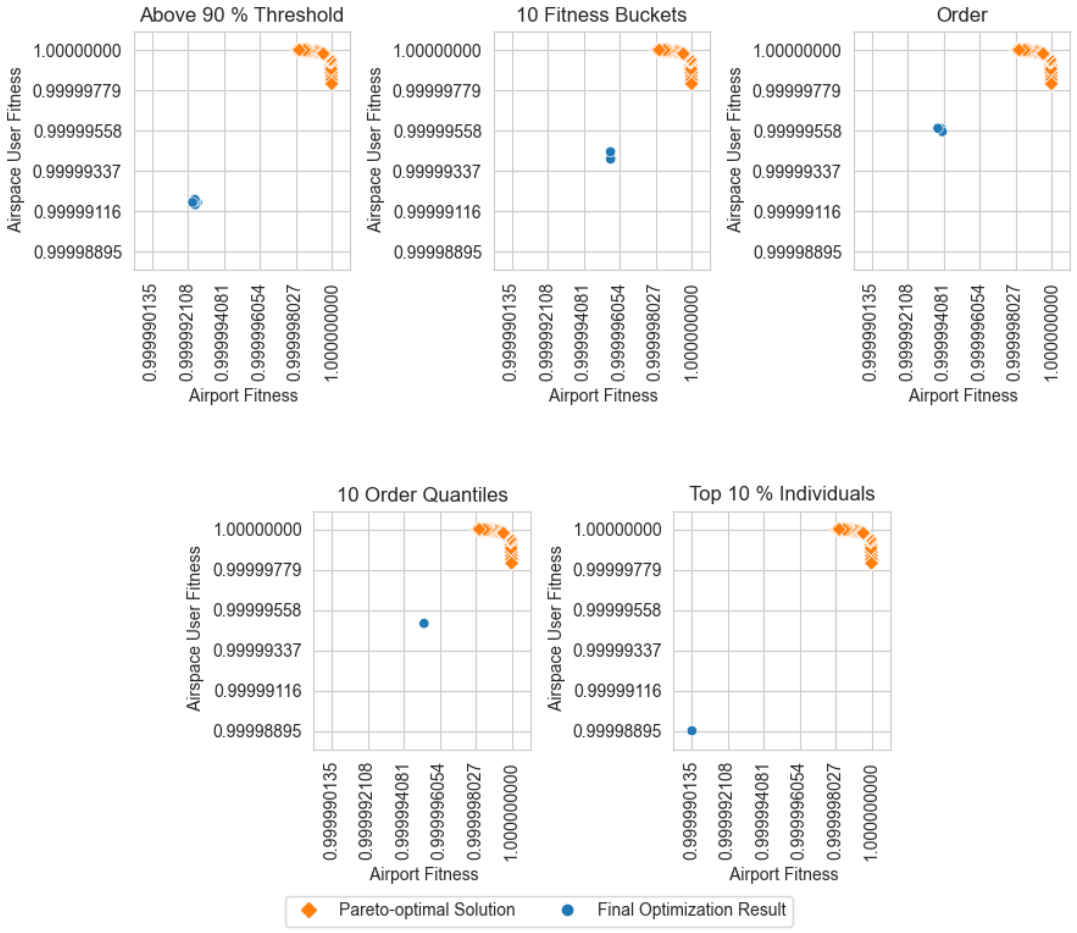

Fig. 2. Optimization results of a sample situation with 122 flights and 233 TTAs

varying number of instances. The runtimes were measured on an Ubuntu system running kernel version 5.15.0-88-generic (#98-Ubuntu), equipped with four Intel Xeon Gold 6248R processors running at 3.00 GHz. As problem size increases, optimization runtime increases within each obfuscation configuration. These results also show the impact of different obfuscation methods on runtime, with *fitness buckets* obfuscation having a strong runtime penalty.

The results in Table V were obtained by running the genetic algorithms for 100 000 evaluations. The Optimizer combined with the Privacy Engine can also be applied in settings with stricter time constraints than the runtimes reported in Table V because genetic algorithms can provide interim results at any point in time. Obfuscation methods with lower runtime overhead enable the genetic algorithms to perform more iterations within a fixed time budget compared to methods with higher overhead, thereby increasing the likelihood of finding high-quality solutions. We provide additional runtime benchmarks for the Privacy Engine online [24].

TABLE III
MEDIAN AND INTERQUARTILE RANGE (IQR) OF THE GENERATIONAL DISTANCE (GD^+) FOR DIFFERENT PROBLEM-SIZE BINS, ROUNDED TO SIX DECIMAL DIGITS

Obfuscation	Problem Size	Instances	Initial Populations		Optimization Results	
			Median	IQR	Median	IQR
Above 90 % Threshold	[3, 73]	353	0.000021	0.000043	0.000000	0.000001
	]73, 143]	230	0.101058	0.064392	0.000003	0.000004
	]143, 214]	352	0.157179	0.046180	0.000005	0.000004
	]214, 284]	61	0.183384	0.033872	0.000006	0.000004
	]284, 355]	21	0.195134	0.041488	0.000007	0.000002
10 Fitness Buckets	[3, 73]	353	0.000019	0.000039	0.000000	0.000000
	]73, 143]	230	0.088444	0.064062	0.000002	0.000003
	]143, 214]	352	0.145088	0.051844	0.000003	0.000003
	]214, 284]	61	0.174419	0.036259	0.000003	0.000002
	]284, 355]	21	0.194515	0.029121	0.000006	0.000001
Order	[3, 73]	353	0.000020	0.000039	0.000000	0.000000
	]73, 143]	230	0.088444	0.064061	0.000001	0.000002
	]143, 214]	352	0.145088	0.051844	0.000002	0.000003
	]214, 284]	61	0.174419	0.036259	0.000003	0.000002
	]284, 355]	21	0.194515	0.029121	0.000006	0.000002
10 Order Quantiles	[3, 73]	353	0.000019	0.000039	0.000000	0.000000
	]73, 143]	230	0.088444	0.064062	0.000001	0.000003
	]143, 214]	352	0.145088	0.051844	0.000003	0.000003
	]214, 284]	61	0.174419	0.036259	0.000003	0.000003
	]284, 355]	21	0.194515	0.029121	0.000006	0.000003
Top 10 % Individuals	[3, 73]	353	0.000020	0.000039	0.000000	0.000000
	]73, 143]	230	0.088444	0.064062	0.000003	0.000004
	]143, 214]	352	0.145088	0.051844	0.000005	0.000004
	]214, 284]	61	0.174419	0.036259	0.000007	0.000004
	]284, 355]	21	0.194515	0.029121	0.000009	0.000004

TABLE IV
MEDIAN AND INTERQUARTILE RANGE (IQR) OF THE INVERTED GENERATIONAL DISTANCE (IGD^+) FOR DIFFERENT PROBLEM-SIZE BINS, ROUNDED TO SIX DECIMAL DIGITS

Obfuscation	Problem Size	Instances	Initial Populations		Optimization Results	
			Median	IQR	Median	IQR
Above 90 % Threshold	[3, 73]	353	0.000029	0.000047	0.000000	0.000003
	]73, 143]	230	0.101058	0.064392	0.000004	0.000004
	]143, 214]	352	0.157180	0.046180	0.000006	0.000004
	]214, 284]	61	0.183385	0.033872	0.000007	0.000003
	]284, 355]	21	0.195134	0.041488	0.000008	0.000002
10 Fitness Buckets	[3, 73]	353	0.000026	0.000043	0.000000	0.000002
	]73, 143]	230	0.088444	0.064062	0.000003	0.000003
	]143, 214]	352	0.145089	0.051844	0.000004	0.000004
	]214, 284]	61	0.174420	0.036259	0.000004	0.000002
	]284, 355]	21	0.194516	0.029121	0.000007	0.000001
Order	[3, 73]	353	0.000019	0.000040	0.000000	0.000000
	]73, 143]	230	0.088444	0.064062	0.000001	0.000002
	]143, 214]	352	0.145089	0.051844	0.000003	0.000003
	]214, 284]	61	0.174420	0.036259	0.000004	0.000003
	]284, 355]	21	0.194516	0.029121	0.000007	0.000002
10 Order Quantiles	[3, 73]	353	0.000022	0.000042	0.000000	0.000001
	]73, 143]	230	0.088444	0.064062	0.000002	0.000003
	]143, 214]	352	0.145089	0.051844	0.000004	0.000003
	]214, 284]	61	0.174420	0.036259	0.000004	0.000003
	]284, 355]	21	0.194516	0.029121	0.000007	0.000003
Top 10 % Individuals	[3, 73]	353	0.000026	0.000044	0.000000	0.000002
	]73, 143]	230	0.088444	0.064062	0.000004	0.000004
	]143, 214]	352	0.145089	0.051844	0.000006	0.000004
	]214, 284]	61	0.174420	0.036259	0.000008	0.000004
	]284, 355]	21	0.194516	0.029121	0.000010	0.000004

TABLE V
MINIMUM, MEDIAN, MAXIMUM AND INTERQUARTILE RANGE (IQR) OF THE OPTIMIZATION RUNTIME IN SECONDS FOR DIFFERENT PROBLEM-SIZE BINS

Obfuscation	Problem Size	Instances	Min	Median	Max	IQR
Above 90 % Threshold	[55, 155]	18	577	624.5	685	62.50
	]155, 255]	29	615	673.0	750	35.00
	]255, 355]	4	705	758.5	787	27.25
10 Fitness Buckets	[55, 155]	18	955	996.0	1 063	57.75
	]155, 255]	29	982	1 052.0	1 098	51.00
	]255, 355]	4	1 092	1 170.0	1 180	31.00
Order	[55, 155]	18	370	439.5	495	30.00
	]155, 255]	29	439	467.0	503	26.00
	]255, 355]	4	518	594.5	609	34.00
10 Order Quantiles	[55, 155]	18	607	691.5	744	70.25
	]155, 255]	29	666	728.0	822	70.00
	]255, 355]	4	772	830.0	873	40.25
Top 10 % Individuals	[55, 155]	18	391	446.5	535	85.00
	]155, 255]	29	413	543.0	592	66.00
	]255, 355]	4	583	594.5	613	17.25

VI. DISCUSSION AND CONCLUSION

The TTMS live trials at Zurich Airport, conducted with a fully trusted platform provider, indicate preliminary improvements across multiple key performance indicators [1]. For SWISS operations, arrival punctuality improved by 20.2 %, while the number of flights experiencing heavy delays decreased by 23.5 % [1]. From the airport's perspective, the number of flights arriving more than 15 minutes late decreased by 18 %, while knock-on delays decreased by 9 % [1]. On the passenger side, overall delay minutes dropped by 11 %, alongside a 14 % reduction in passengers experiencing missed connections and resulting follow-on delays [1]. For SWISS passengers, the number of passengers experiencing missed connections decreased by 33.3 %, while those with critical connections decreased by 19.8 % [1].

In this paper, we investigated the practicability of extending the TTMS for scenarios in which the platform provider is not fully trusted. We considered a privacy-preserving TTMS that combines evolutionary algorithms and MPC to protect the confidential input provided by the AUs from the platform provider. We evaluated the impact of privacy-preserving computation on the quality of flight lists and optimization runtime using real-world data from the TTMS live trials at Zurich Airport. The experimental results demonstrate that such a privacy-preserving TTMS is able to find solutions almost as good as Pareto-optimal solutions, while runtime measurements demonstrate that the TTMS is able to find solutions within practical time constraints. The privacy-preserving TTMS is a promising alternative for scenarios in which AUs do not fully trust the platform provider.

The primary goal of the experiments was the investigation of the trade-off between privacy-preserving computation and quality of the solution found by the Optimizer. GD^+ and IGD^+ values between the final optimization results and the Pareto-optimal solutions allow for such investigation, but these metrics do not directly translate to impacts on key performance indicators. Quantifying the impact of privacy-preserving computation on operational metrics would require additional data, which were not available in our lab environment; such an investigation remains for future work.

The interpretation of the experimental results is difficult due to very large negative weights associated with infeasible assignments. The largest optimization instance, comprising 122 flights and 233 TTAs, exhibits a maximum fitness of 9 157 239 for the airport preferences and 7 007 791 for AU preferences, while the minimum fitness values are $-208\,308\,295\,815$ and $-208\,308\,413\,356$, respectively. For instance, one of the solutions obtained from the final optimization result with *order* obfuscation yields fitness values of 7 896 520 for airport preferences and 6 110 613 for AU preferences, corresponding to normalized fitness values of approximately 0.9999939 and 0.9999957, respectively. While normalized fitness values enable aggregation of GD^+ and IGD^+ across optimizations and regulations, they are less intuitive to interpret than absolute fitness values.

In our experiments, the evolutionary algorithm always started from a population of randomly generated individuals. Optimization performance may be further improved if the Optimizer has access to the original flight list. Future research could investigate alternative evolutionary algorithms as well as local search approaches in scenarios where the original flight list is available.

While short-term inequities between AUs may be acceptable to improve overall efficiency, no single AU must be consistently advantaged or disadvantaged compared to others over multiple regulations. Long-term equity between AUs can be achieved by modifying weights based on the results of previous optimizations [12]. Harzfeld et al. [29] propose different mechanisms for calculating such weight modifications using the Theil index as a measure for inequity. In general, for each AU, to determine the weight modification, the AU's contribution to the Theil index is calculated for a rolling window of 20 regulations. An AU's calculated contribution to the Theil index then serves to modify that AU's weights; the

modification depends on the mechanism. Harzfeld et al. [29] evaluate the different mechanisms using data from the TTMS live trials at Zurich Airport. The results show that the Theil index can be reduced with only minor impact on the quality of the flight lists. Future work could investigate these mechanisms with a privacy-preserving TTMS.

Acknowledgments

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) and Grammarly (Superhuman Platform Inc.) for language editing to improve readability and clarity. After using these tools, the authors carefully reviewed and revised the content as necessary and take full responsibility for the final version of the manuscript.

This work was conducted as part of the HARMONIC project. This project has received funding from the SESAR Joint Undertaking under grant agreement No 101114675 under the European Union's Horizon Europe research and innovation program. UK participant NATS in Horizon Europe Project HARMONIC receives funding from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10091990]. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI). The views expressed in this paper are those of the authors.

Co-funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

References

[1] L.-M. Hagemann, M. Carré, M. Brügger, G. Sarrazin, and A. Lacroix, "Multi-stakeholder Optimized Arrival Management with the Target Time Management System (TTMS)," in *Proceedings of the 15th SESAR Innovation Days*, 2025.

[2] N. Pilon, A. Cook, S. Ruiz, A. Bujor, and L. Castelli, "Improved flexibility and equity for airspace users during demand-capacity imbalance-an introduction to the user-driven prioritisation process," in *Proceedings of the 6th SESAR Innovation Days*, 2016.

[3] N. Pilon, L. Guichard, Z. Bazso, G. Murgese, and M. Carré, "User-driven prioritisation process (udpp) from advanced experimental to pre-operational validation environment," *Journal of Air Transport Management*, vol. 97, p. 102124, 2021.

[4] European Union, "A Privacy-Preserving Marketplace for Slot Management," https://doi.org/10.3030/890456, 2020.

[5] C. G. Schuetz, T. Lorünser, S. Jaburek, K. Schuetz, F. Wohner, R. Karl, and E. Gringinger, "A distributed architecture for privacy-preserving optimization using genetic algorithms and multi-party computation," in *Cooperative Information Systems*, M. Sellami, P. Ceravolo, H. A. Reijers, W. Gaaloul, and H. Panetto, Eds. Cham: Springer International Publishing, 2022, pp. 168–185.

[6] E. Gringinger, S. Ruiz, and C. G. Schuetz, "Business and economic concepts for a privacy-preserving marketplace for atfm slots," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. IEEE, 2022, pp. 1–11.

[7] C. G. Schuetz, E. Gringinger, N. Pilon, and T. Lorünser, "A privacy-preserving marketplace for air traffic flow management slot configuration," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–9.

[8] European Union, "HARMONIsed network through smart technology and Collaboration," https://doi.org/10.3030/101114675, 2023.

[9] S. Gruber, P. Feichtenschlager, C. Fabianek, E. Gringinger, and C. G. Schuetz, "Towards a heuristic optimizer for a target time management system in air traffic flow management," in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, 2024, pp. 1–10.

[10] S. Gruber, P. Feichtenschlager, and C. G. Schuetz, "Using genetic algorithms for privacy-preserving optimization of multi-objective assignment problems in time-critical settings: An application in air traffic flow management," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2024, Melbourne, VIC, Australia, July 14-18, 2024*, X. Li and J. Handl, Eds. ACM, 2024.

[11] T. Harzfeld, S. Gruber, C. G. Schuetz, M. Carré, M. Brügger, A. Lacroix, C. Fabianek, C. Rihacek, and E. Gringinger, "Towards conformance criteria for ensuring fairness among airspace users in collaborative optimization of flight lists in air traffic flow management," in *2025 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2025, pp. 1–14.

[12] M. Carré, "Multi-airline operations optimization under major disruptions," Ph.D. dissertation, Université Clermont Auvergne, 2024. [Online]. Available: https://theses.hal.science/tel-04547756

[13] T. Lorünser, F. Wohner, and S. Krenn, "A verifiable multiparty computation solver for the linear assignment problem: And applications to air traffic management," in *Proceedings of the 2022 on Cloud Computing Security Workshop*, 2022, pp. 41–51.

[14] B. Zhao, W. Chen, F. Wei, X. Liu, Q. Pei, and J. Zhang, "PEGA: A privacy-preserving genetic algorithm for combinatorial optimization," *IEEE Trans. Cybern.*, vol. 54, no. 6, pp. 3638–3651, 2024.

[15] D. Funke and F. Kerschbaum, "Privacy-preserving multi-objective evolutionary algorithms," in *Parallel Problem Solving from Nature, PPSN XI*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 41–50.

[16] J. Sakuma and S. Kobayashi, "A genetic algorithm for privacy preserving combinatorial optimization," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 1372–1379.

[17] S. Han and W. K. Ng, "Privacy-preserving genetic algorithms for rule discovery," in *Data Warehousing and Knowledge Discovery*, I. Y. Song, J. Eder, and T. M. Nguyen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 407–417.

[18] K. Schuetz, C. G. Schuetz, and S. Jaburek, "Privacy-preserving implementation of local search algorithms for collaboratively solving assignment problems in time-critical contexts," in *2023 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2023, pp. 1–10.

[19] S. Gruber, T. Harzfeld, and C. G. Schuetz, "HARMONIC-Optimizer-Python," https://doi.org/10.5281/zenodo.20729054, 2026.

[20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[21] A. F. Gad, "PyGAD: an intuitive genetic algorithm python library," *Multim. Tools Appl.*, vol. 83, no. 20, pp. 58 029–58 042, 2024.

[22] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[23] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.

[24] F. Wohner, "ait-crypto/privacy-engine: v1.0.0," https://doi.org/10.5281/zenodo.20748035, 2026.

[25] M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1575–1590.

[26] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 805–817.

[27] R. Kruse, S. Mostaghim, C. Borgelt, C. Braune, and M. Steinbrecher, *Fundamental Evolutionary Algorithms*. Springer, 2022, pp. 287–341.

[28] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *EMO 2015 – Part II*, ser. LNCS, A. Gaspar-Cunha, C. H. Antunes, and C. A. C. Coello, Eds., vol. 9019. Springer, 2015, pp. 110–125.

[29] T. Harzfeld, S. Gruber, C. G. Schuetz, C. Fabianek, C. Rihacek, and E. Gringinger, "Towards a Mechanism for Ensuring Equity Over Time Among Airspace Users in Collaborative Flight Prioritization," in *2026 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2026, to appear.